



Calhoun: The NPS Institutional Archive

Faculty and Researcher Publications

Faculty and Researcher Publications

2010-04-12

Adversarial TCP: An Offensive TCP Stack to Penalize Abusive Connections (Poster)

Craven, Ryan

Monterey, California. Naval Postgraduate School

<http://hdl.handle.net/10945/37788>



Calhoun is a project of the Dudley Knox Library at NPS, furthering the precepts and goals of open government and government transparency. All information contained herein has been approved for release by the NPS Public Affairs Officer.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

<http://www.nps.edu/library>

Adversarial TCP: An Offensive TCP Stack to Penalize Abusive Connections

Ryan Craven, Kristina Foster, Robert Beverly {rcraven, kmfoster, rbeverly}@nps.edu

Motivation

Penalize abusive hosts, spam bots, DoS attacks, scam infrastructure, etc. Cause suspected abusive connections to:

- Send more traffic
- Consume more bandwidth / time
- Induce more congestion
- Be more visible (bandwidth, congestion, \$\$, etc.)

Prior Work

- TCP “tarpits” to artificially slow abusive connections (we aim to do the opposite)
- Exploiting traffic congestion characteristics of abusive hosts (often bots with asymmetric bandwidth)

Hypothesis

An “adversarial” TCP stack (A-TCP) can cause a remote TCP to perform more work.

Questions

Initial research highlights interesting questions:

- How to induce extra work?
- Metric of work: packets, bytes, time, etc.?
- Ratio of extra work performed by A-TCP versus induced remote work?
- Differences in A-TCP’s effects against various operating systems?
- Can abusive hosts distinguish between normal and A-TCP?

Approach 1: TCP MSS

- **Idea:** reduce advertised maximum segment size (MSS)
- Abusive host sends more packets with less data per packet = higher header overhead
- Higher header overhead = more work
- Hook TCP via iptables NFQUEUE bindings
- Scapy script overwrites MSS in SYN-ACK



Experiment

- Isolated test-bed with real hardware, different OS, dummynet, etc.
- 60 runs of 8MB transfer at different A-TCP MSS
- Different A-TCP loss rates to trigger fast-retransmit
- Define “Asynchronous Payoff Ratio” (APR):

$S_{TCP}(N)$ = TCP bytes xmit’d to send N byte data

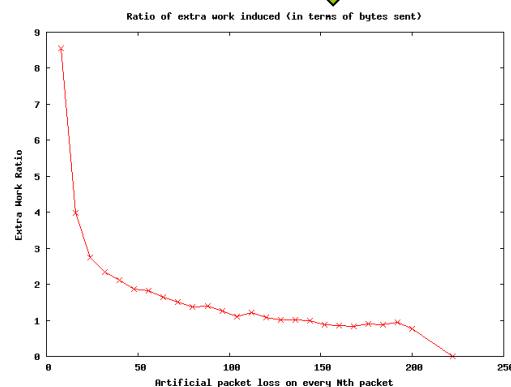
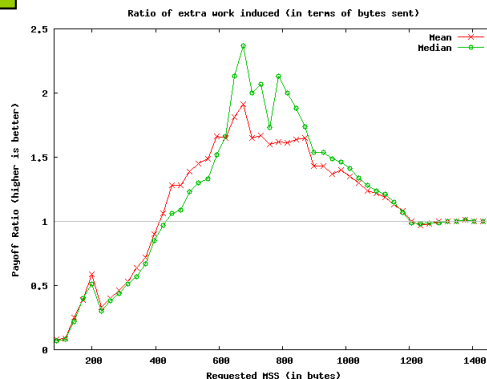
$R_{TCP}(N)$ = TCP bytes xmit’d to receive N byte data

APR =

$$\frac{S_{ATCP}(N) - S_{TCP}(N)}{R_{ATCP}(N) - R_{TCP}(N)} = \frac{\text{Attacker extra bytes}}{\text{A-TCP extra bytes}}$$

Approach 2: RFC2581

- **Idea:** fake loss and induce remote side fast-retransmit / fast-recovery
- Abusive host must retransmit lost data or entire outstanding window = work
- Challenge is to prevent remote TCP from collapsing congestion window
- Remote TCP cannot differentiate real packet loss from A-TCP’s artificial loss



Early Results

- Significant OS differences (e.g. Win7 MSS)
- Large feasible MSS range with APR > 2
- MSS < 400 requires extra ACKs leading to APR < 1
- A-TCP artificial loss + fast retransmit can produce large APR – challenge is congestion window
- We believe order of magnitude higher APRs possible – subject of our current research